Bahan Kuliah II4021 Kriptografi

Elgamal Signature Scheme, DSA, ECDSA, dan Schnorr Signature Scheme

Oleh: Rinaldi M

Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
ITB - 2026

Sebagian materi di dalam PPT ini diambil dari sumber:

1. Fatimah Al-Ubaidy, *Digital Signature*, in course *Data Security*, Mustansiriyah University, Faculty of Engineering, Computer Engineering Dept.
2. Chirag's Blog, Elgamal and Schnorr scheme of Digital Signature | Which scheme is best Elgamal or Schnorr?, https://www.chiragbhalodia.com/2021/11/elgamal-schnorr-scheme.html

ELGAMAL DIGITAL SIGNATURE SCHEME

Elgamal encryption scheme is designed to enable encryption by a user's public key with decryption by the user's private key. The Elgamal signature scheme involves the use of the private key for digital signature generation and the public key for digital signature verification.

Preliminary:

If q is a prime number and α is a primitive root of q, then

1. For any integer m, $\alpha^m \equiv 1 \pmod{q}$ if and only if $m \equiv 0 \pmod{q-1}$.
2. For any integers, i, j, $\alpha^i \equiv \alpha^j \pmod{q}$ if and only if $i \equiv j \pmod{q-1}$.

Elgamal Key Generation:

As with Elgamal encryption, the global elements of **Elgamal digital signature** are a prime number ***q*** and **α**, which is a primitive root of ***q***. User **A** generates a private/public key pair as follows:

1. Generate a random integer X_A, such that $1 < X_A < q - 1$.
2. Compute $Y_A = \alpha^{X_A} \bmod q$.
3. A's private key is X_A; A's pubic key is $\{q, \alpha, Y_A\}$.

Elgamal Digital Signature Generation:

To sign a message ***M***, user **A** first computes the hash ***m* = H(*M*)**, such that ***m*** is an integer in the range **$0 \leq m \leq q - 1$**. **A** then forms a digital signature as follows:

Sumber: Fatimah Al-Ubaidy, Digital Signature

Elgamal Digital Signature Generation (Continued):

1. Choose a random integer K such that $1 \leq K \leq q - 1$ and $\gcd(K, q - 1) = 1$. That is, K is relatively prime to $q - 1$.
2. Compute $S_1 = \alpha^K \bmod q$. Note that this is the same as the computation of C_1 for Elgamal encryption.
3. Compute $K^{-1} \bmod (q - 1)$. That is, compute the inverse of K modulo $q - 1$.
4. Compute $S_2 = K^{-1}(m - X_A S_1) \bmod (q - 1)$.
5. The signature consists of the pair (S_1, S_2).

Elgamal Digital Signature Verification:

Any user **B** can verify the signature as follows:

1. Compute $V_1 = \alpha^m \bmod q$.
2. Compute $V_2 = (Y_A)^{S_1}(S_1)^{S_2} \bmod q$.

The signature is valid if $\boldsymbol{V_1} = \boldsymbol{V_2}$. Let us demonstrate that this is so.

Example (1):

Let us start with the prime field GF(19); that is, $\boldsymbol{q}$ = 19. It has primitive roots {2, 3, 10, 13, 14, 15}. We choose $\boldsymbol{\alpha}$ = 10.

Alice generates a key pair as follows:

1. Alice chooses $X_A = 16$.
2. Then $Y_A = \alpha^{X_A} \bmod q = \alpha^{16} \bmod 19 = 4$.
3. Alice's private key is 16; Alice's pubic key is $\{q, \alpha, Y_A\} = \{19, 10, 4\}$.

Sumber: Fatimah Al-Ubaidy, Digital Signature

Example (1): (Continued)

(Sign) Suppose Alice wants to sign a message with hash value $\boldsymbol{m} = 14$.

1. Alice chooses $K = 5$, which is relatively prime to $q - 1 = 18$.
2. $S_1 = \alpha^K \bmod q = 10^5 \bmod 19 = 3$ (see Table 2.7).
3. $K^{-1} \bmod (q - 1) = 5^{-1} \bmod 18 = 11$.
4. $S_2 = K^{-1}(m - X_A S_1) \bmod (q - 1) = 11\,(14 - (16)(3)) \bmod 18 = -374 \bmod 18 = 4$.

(Verify): Bob can verify the signature as follows:

1. $V_1 = \alpha^m \bmod q = 10^{14} \bmod 19 = 16$.
2. $V_2 = (Y_A)^{S_1}(S_1)^{S_2} \bmod q = (4^3)(3^4) \bmod 19 = 5184 \bmod 19 = 16$.

Thus, the signature is valid because $\boldsymbol{V_1} = \boldsymbol{V_2}$.

NIST Digital Signature Approach:

NIST has published Federal Information Processing Standard **FIPS 186**, known as the Digital Signature Algorithm (**DSA**). The **DSA** makes use of the Secure Hash Algorithm (**SHA**). The **DSA** was originally proposed in 1991. Several expanded versions of the standard were then issued as FIPS 186-2, FIPS 186-3 and FIPS 186-4 in response to public feedback concerning the security of the scheme. This latest version also incorporates digital signature algorithms based on RSA and on elliptic curve cryptography.

Sumber: Fatimah Al-Ubaidy, Digital Signature

Digital Signature Standard

- DSS (*Digital Signature Standard*) adalah bakuan (standard) untuk tanda-tangan digital.
- Diresmikan pada bulan Agustus 1991 oleh NIST (*The National Institute of Standard and Technology*)

- *DSS* terdiri dari dua komponen:

  1. Algoritma tanda-tangan digital: *Digital Signature Algorithm* (*DSA*).

  2. Fungsi *hash* standard: *Secure Hash Algorithm* (*SHA-1*).

     → Sudah dibahas pada materi kuliah fungsi *hash*

Digital Signature Algorithm (*DSA*)

- *DSA* termasuk ke dalam algoritma kriptografi kunci-publik.
- *DSA* tidak dapat digunakan untuk enkripsi pesan; *DSA* dispesifikasikan khusus untuk tanda-tangan digital saja.

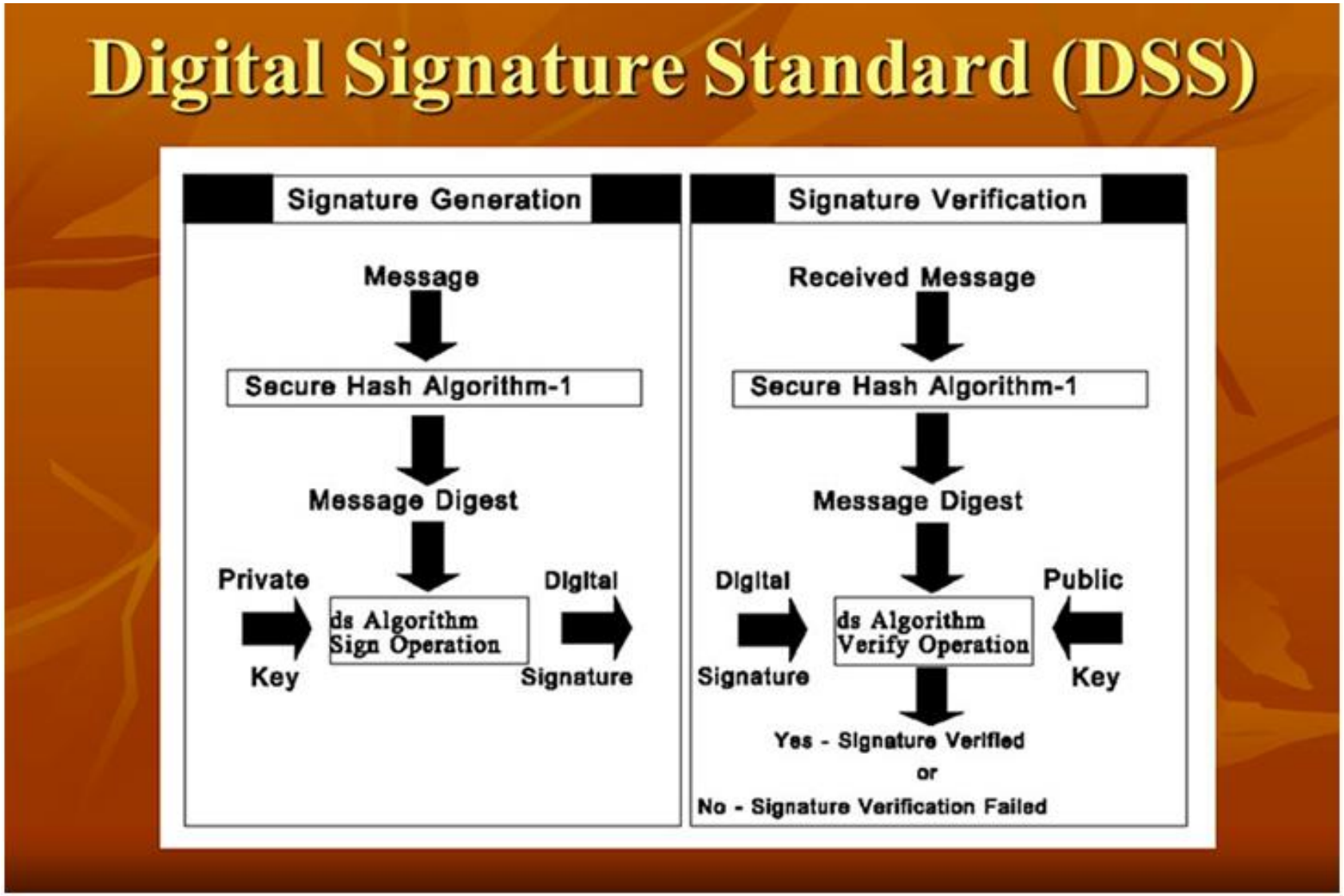
- *DSA* mempunyai dua fungsi utama:
  1. Pembangkitan tanda-tangan (*signature generation*),
  2. Pemeriksaan keabsahan tanda-tangan (*signature verification*).

- *DSA* dikembangkan dari algoritma *ElGamal Signature*.

- *DSA* menggunakan dua buah kunci, yaitu kunci publik dan kunci privat.

- Pembentukan tanda-tangan menggunakan kunci privat, sedangkan verifikasi tanda-tangan menggunakan kunci publik.

- DSA menggunakan fungsi *hash SHA-1* (*Secure Hash Algorithm*) untuk menghasilkan *message digest* yang berukuran 160 bit (SHA- sudah dijelaskan pada materi kuliah sebelumnya).



Sumber: https://signx.wondershare.com/knowledge/digital-signature-algorithm.html

Parameter DSA

1. p, bilangan prima, panjangnya L bit, $512 \leq L \leq 1024$ dan L harus kelipatan 64. Parameter p bersifat publik.
2. q, bilangan prima 160 bit, merupakan faktor dari $p - 1$. Dengan kata lain, $(p - 1) \bmod q = 0$. Parameter q bersifat publik.
3. $g = h^{(p-1)/q} \bmod p$, $h < p - 1$ sedemikian sehingga $h^{(p-1)/q} \bmod p > 1$. Parameter g bersifat publik.
4. x, kunci privat, adalah bilangan bulat kurang dari q.
5. $y = g^x \bmod p$, kunci publik.
6. m, pesan yang akan diberi tanda-tangan.

Pembangkitan Sepasang Kunci

1. Pilih bilangan prima p dan q, yang dalam hal ini $(p - 1) \bmod q = 0$.
2. Hitung $g = h^{(p-1)/q} \bmod p$, yang dalam hal ini $1 < h < p - 1$ dan $h^{(p-1)/q} \bmod p > 1$.
3. Tentukan kunci privat x, yang dalam hal ini x, dalam hal ini $0 < x < q$.
4. Hitung kunci publik $y = g^x \bmod p$.

Prosedur di atas menghasilkan:

parameter publik: (p, q, g, y)

parameter privat: x

Pembangkitan Tanda-tangan (*Signing*)

1. Hitung *message digest* pesan m dengan fungsi *hash* SHA-1, $H(m)$.

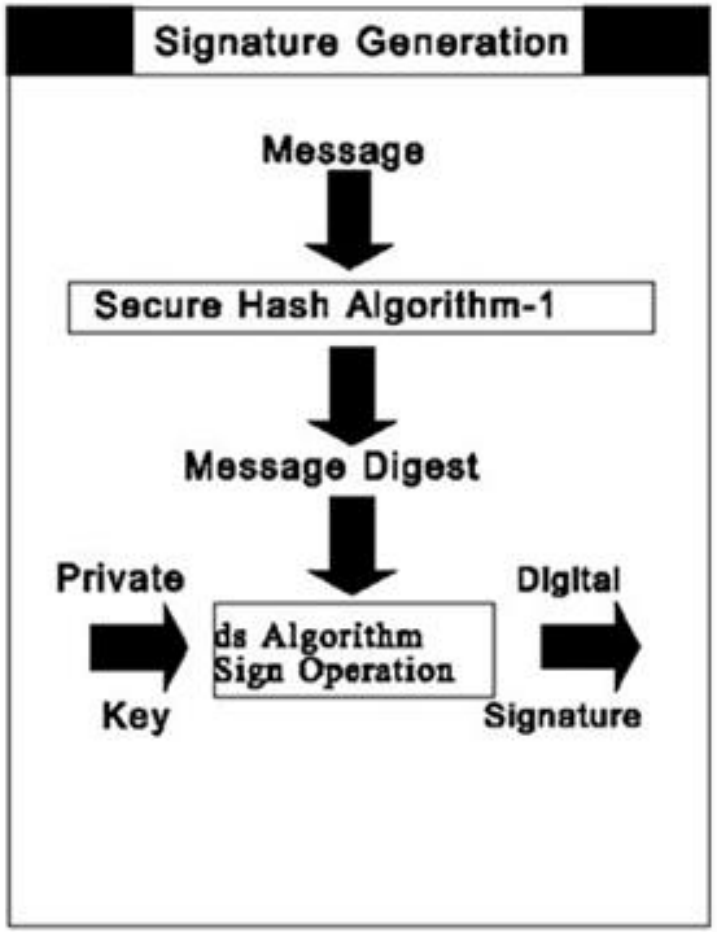
2. Tentukan bilangan acak k, $0 < k < q$.

3. Tanda-tangan dari pesan m adalah bilangan r dan s. Hitung r dan s sebagai berikut (kunci privat = x):

   $$r = (g^k \bmod p) \bmod q$$

   $$s = (k^{-1}(H(m) + x \cdot r)) \bmod q$$

4. Kirim pesan m beserta tanda-tangan (r, s)



Verifikasi Keabsahan Tanda-tangan (*Verifying*)

1. Hitung *message digest* pesan *m* dengan fungsi *hash* SHA-1, $H(m)$.

2. Verifikasi tanda-tangan, *r* dan *s*, sebagai berikut (kunci publik = *y*): :

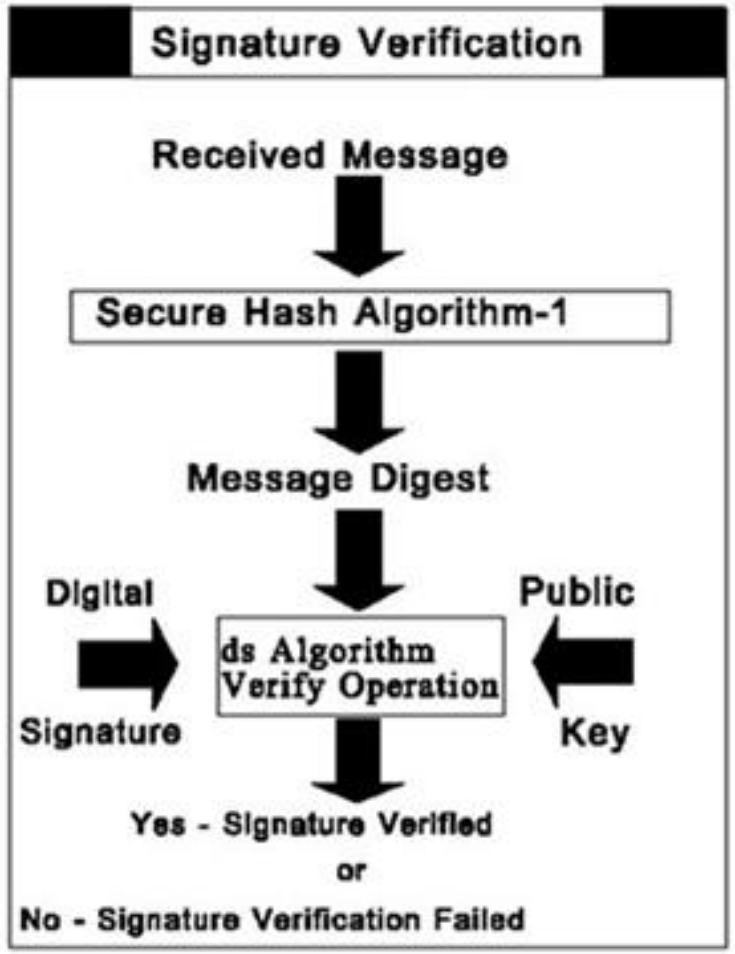
$$w = s^{-1} \bmod q$$

$$u_1 = (H(m) \cdot w) \bmod q$$

$$u_2 = (r \cdot w) \bmod q$$

$$v = ((g^{u1} \cdot y^{u2}) \bmod p) \bmod q)$$

2. Jika $v = r$, maka tanda-tangan digital sah (terverifikasi), sebaliknya tidak sah.



Ringkasan DSA

Global Public-Key Components

p prime number where $2^{L-1} < p < 2^L$ for $512 \leq L \leq 1024$ and L a multiple of 64; i.e., bit length of between 512 and 1024 bits in increments of 64 bits

q prime divisor of $(p - 1)$, where $2^{159} < q < 2^{160}$; i.e., bit length of 160 bits

g $= h^{(p-1)/q} \bmod p$, where h is any integer with $1 < h < (p - 1)$ such that $h^{(p-1)/q} \bmod p > 1$

User's Private Key

x random or pseudorandom integer with $0 < x < q$

User's Public Key

$y = g^x \bmod p$

User's Per-Message Secret Number

k = random or pseudorandom integer with $0 < k < q$

Signing

$r = (g^k \bmod p) \bmod q$

$s = [k^{-1} (\mathrm{H(M)} + xr)] \bmod q$

Signature = (r, s)

Verifying

$w = (s')^{-1} \bmod q$

$u_1 = [\mathrm{H(M')}w] \bmod q$

$u_2 = (r')w \bmod q$

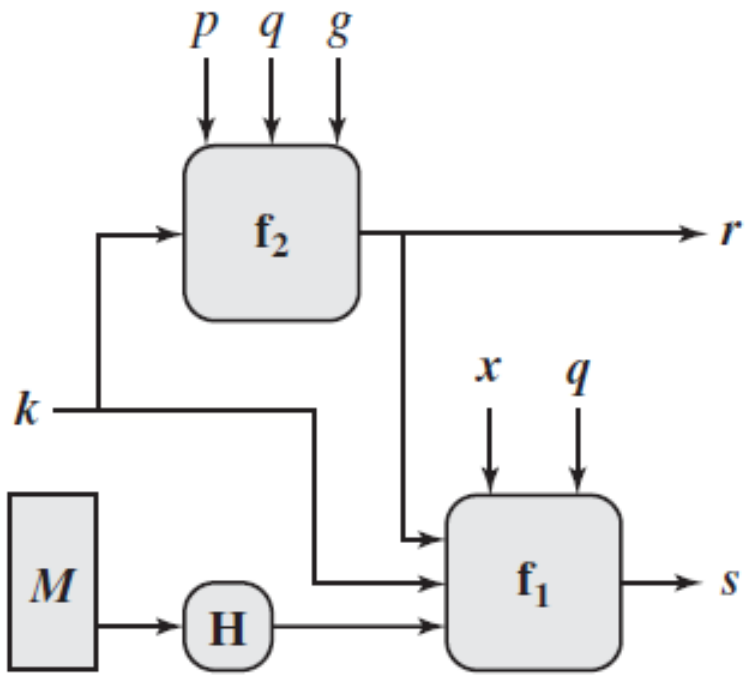
$v = [(g^{u1} y^{u2}) \bmod p] \bmod q$

TEST: $v = r'$

M = message to be signed

$\mathrm{H}(M)$ = hash of M using SHA-1

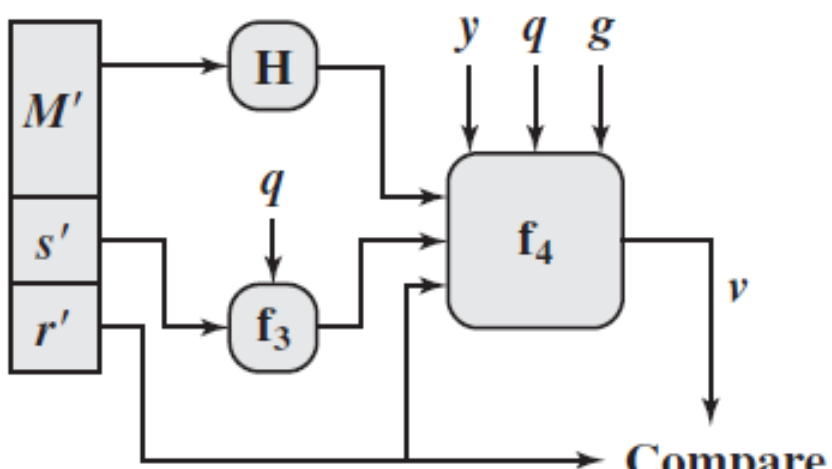
M', r', s' = received versions of M, r, s



$s = \mathrm{f}_1(\mathrm{H}(M), k, x, r, q) = (\mathrm{k}^{-1}\,(\mathrm{H}(M) + xr)) \bmod q$

$r = \mathrm{f}_2(k, p, q, g) = (g^k \bmod p) \bmod q$

(a) Signing



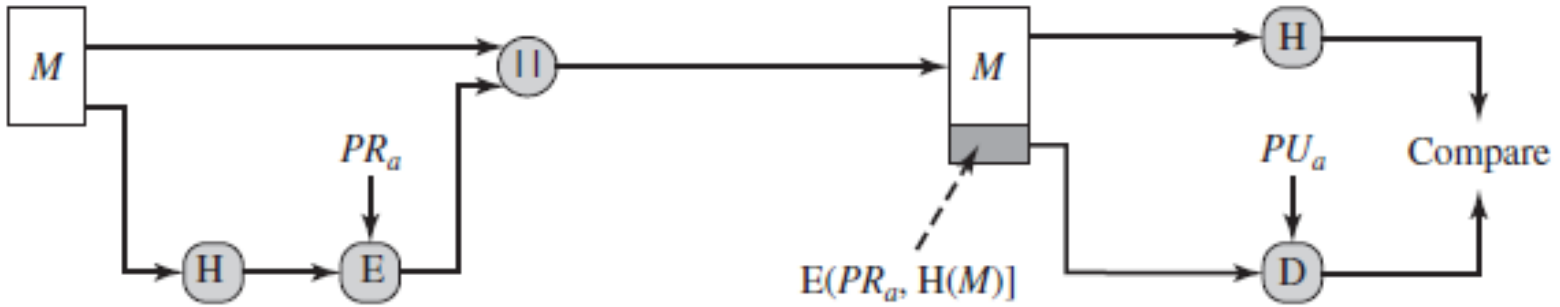
$w = \mathrm{f}_3(s', q) = (s')^{-1} \bmod q$

$v = \mathrm{f}_4(\mathrm{y}, q, g, \mathrm{H}(M'), w, r')$

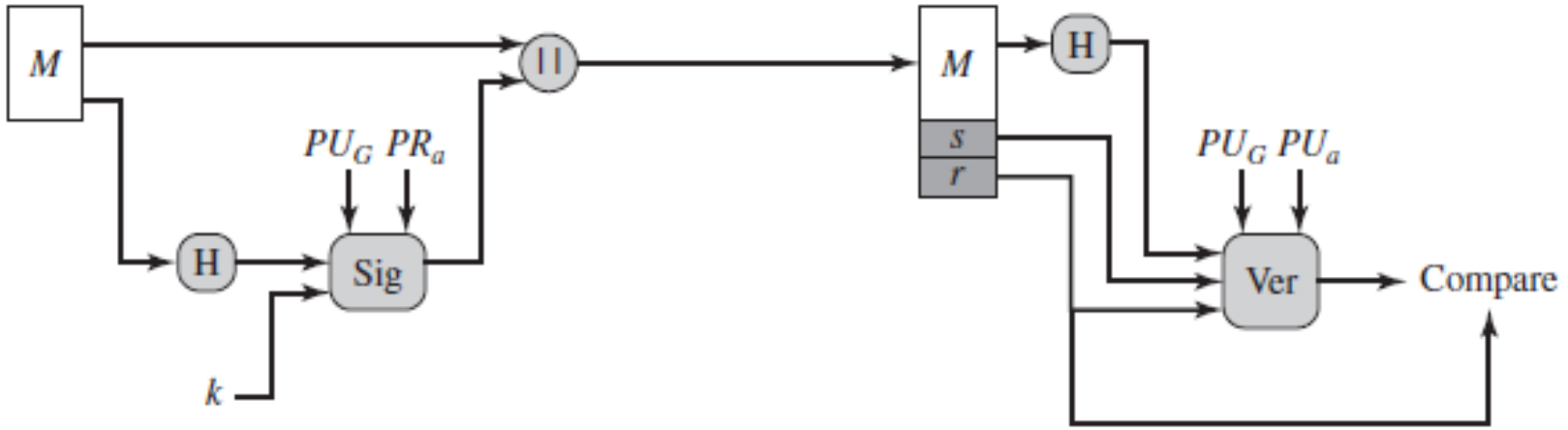
$= ((g^{(\mathrm{H}(M')w) \bmod q}\, \mathrm{y}^{r'w \bmod q}) \bmod p) \bmod q$

(b) Verifying

Perbandingan DSA dengan RSA dalam tanda tangan digital



(a) RSA approach



(b) DSS approach

Contoh Perhitungan *DSA*

A. Prosedur Pembangkitan Sepasang Kunci

1. Pilih bilangan prima p dan q, yang dalam hal ini $(p - 1) \bmod q = 0$.

   $p = 59419$

   $q = 3301$ (memenuhi $(59419 - 1) \bmod 3301 = 0$)

2. Hitung $g = h^{(p-1)/q} \bmod p$, yang dalam hal ini $1 < h < p - 1$ dan $h^{(p-1)/q} \bmod p > 1$.

   $g = 100^{(59419-1)/3301} \bmod (59419) = 18870$ (dengan $h = 100$)

3. Tentukan kunci privat x, yang dalam hal ini $x < q$.

   $x = 3223$

4. Hitung kunci publik $y = g^x \bmod p$.

   $y = 18870^{3223} \bmod 59419 = 29245$ (cek dengan Wolframalpha ☺)

B. Prosedur Pembangkitan Tanda-tangan (Signing)

1. Hitung nilai *hash* dari pesan m, misalkan $H(m) = 4321$

2. Tentukan bilangan acak $k < q$.

   $k = 997$

   $k^{-1} \equiv 2907 \pmod{3301}$

parameter publik: ($p = 59419$, $q = 3301$, $g = 18870$)
parameter privat: $x = 3223$

3. Hitung tanda-tangan digital, r dan s, sebagai berikut:

   $r = (g^k \bmod p) \bmod q = (18870^{997} \bmod 3301) = 848$

   $s = (k^{-1}(H(m) + x \cdot r)) \bmod q = (2907\,(4321 + 3223 \cdot 848)) \bmod 3301$

   $= 7957694475 \bmod 3301 = 183$

4. Kirim pesan m dan tanda-tangan, $(r, s) = (848, 183)$

C. Prosedur Verifikasi Tanda-tangan

1. Hitung nilai *hash* dari pesan *m*, misalkan $H(m) = 4321$
2. Verifikasi tanda-tangan, $(r, s) = (848, 183)$, sebagai berikut:

parameter publik: ($p = 59419$, $q = 3301$, $g = 18870$, $y = 29245$)

$s^{-1} \equiv 469 \pmod{3301}$

$w = s^{-1} \bmod q = 469 \bmod 3301 = 469$

$u_1 = (H(m) \cdot w) \bmod q = (4321 \cdot 469)\ 2026549 \bmod 3301 = 3036$

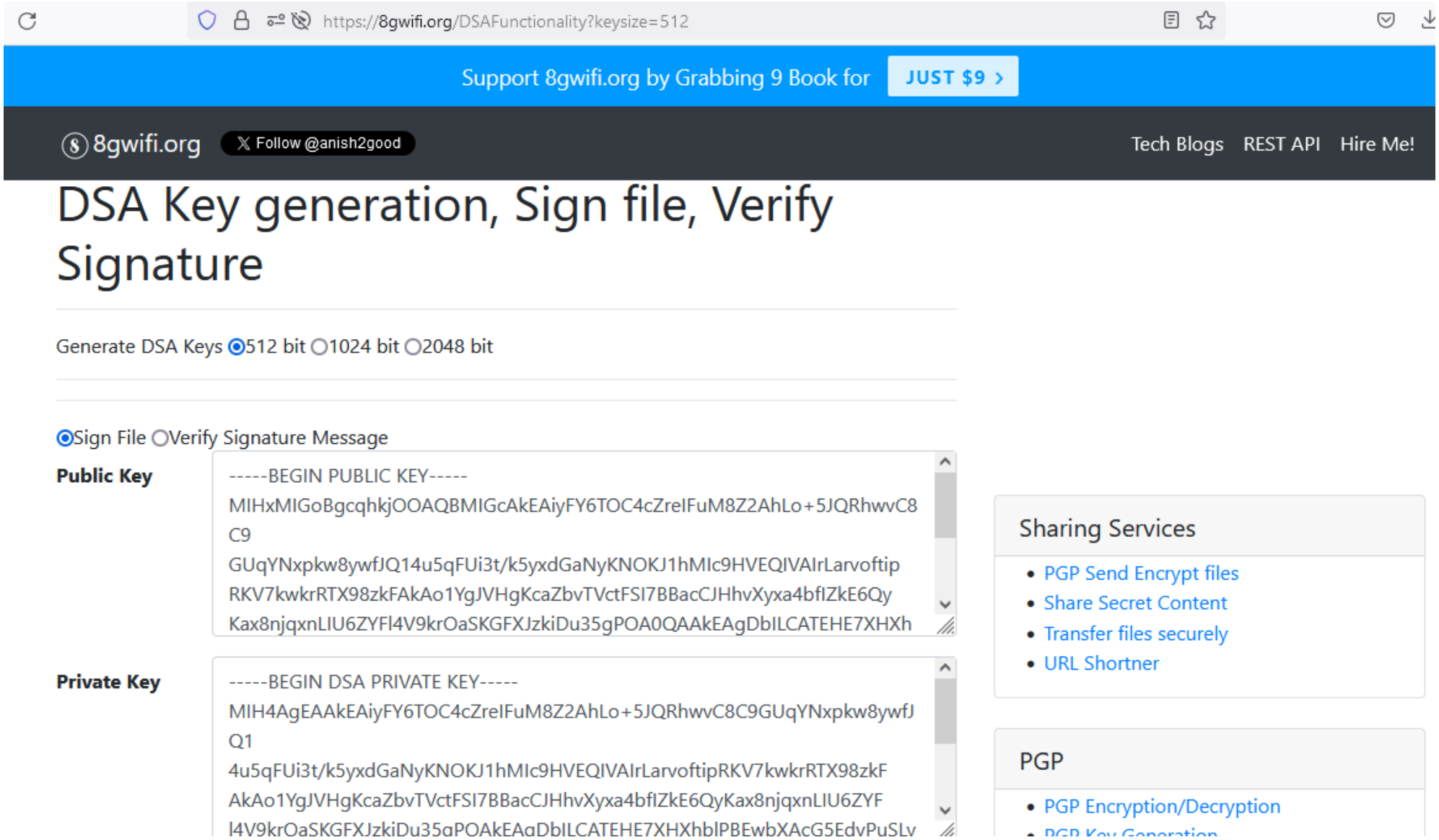
$u_2 = (r \cdot w) \bmod q = (848 \cdot 469) = 397712 \bmod 3301 = 1592$

$v = ((g^{u1} \cdot y^{u2}) \bmod p) \bmod q) = (18870^{3086} \cdot 29245^{1592}) \bmod 3301$

$= 3036\ 848 \bmod 3301 = 848$

3. Karena $v = r$, maka tanda-tangan sah.

Kalkulator DSA online:
https://8gwifi.org/DSAFunctionality?keysize=512



ECDSA

- ECDSA adalah varian DSA untuk komputasi dalam kurva elliptic, sehingga diberi nama *Elliptic Curve Digital Signature Algorithm*

- Besaran yang digunakan di dalam ECDSA
    1. Persamaan kurva elliptic $y^2 = x^3 + ax + b \pmod p$
    2. Titik basis G (dipilih dari himpunan titik di dalam kurva *elliptic*)
    3. Bilangan bulat n, dengan syarat nG = O, O adalah titik di *infinity*
    4. Kunci privat pengirim pesan, bilangan bulat d
    5. Kunci publik pengirim pesan, titik Q = dG
    6. Pesan, m

1. Pembangkitan tanda tangan digital

1. Pilih bilangan acak k, $1 \leq k \leq n - 1$
2. Hitung $kG = (x_1, y_1)$
3. Hitung $r = x_1 \bmod p$
4. Hitung $k^{-1} \bmod q$
5. Hitung e = HASH(m), m adalah pesan yang akan ditandatangani, HASH adalahfungsi hash yang digunakan, misalmya SHA-1, SHA-2, dsb.
6. Hitung $s = k^{-1} (e + dr) \bmod p$, d adalah kunci privat pengirim
7. Tanda-tangan digital adalah (r, s)

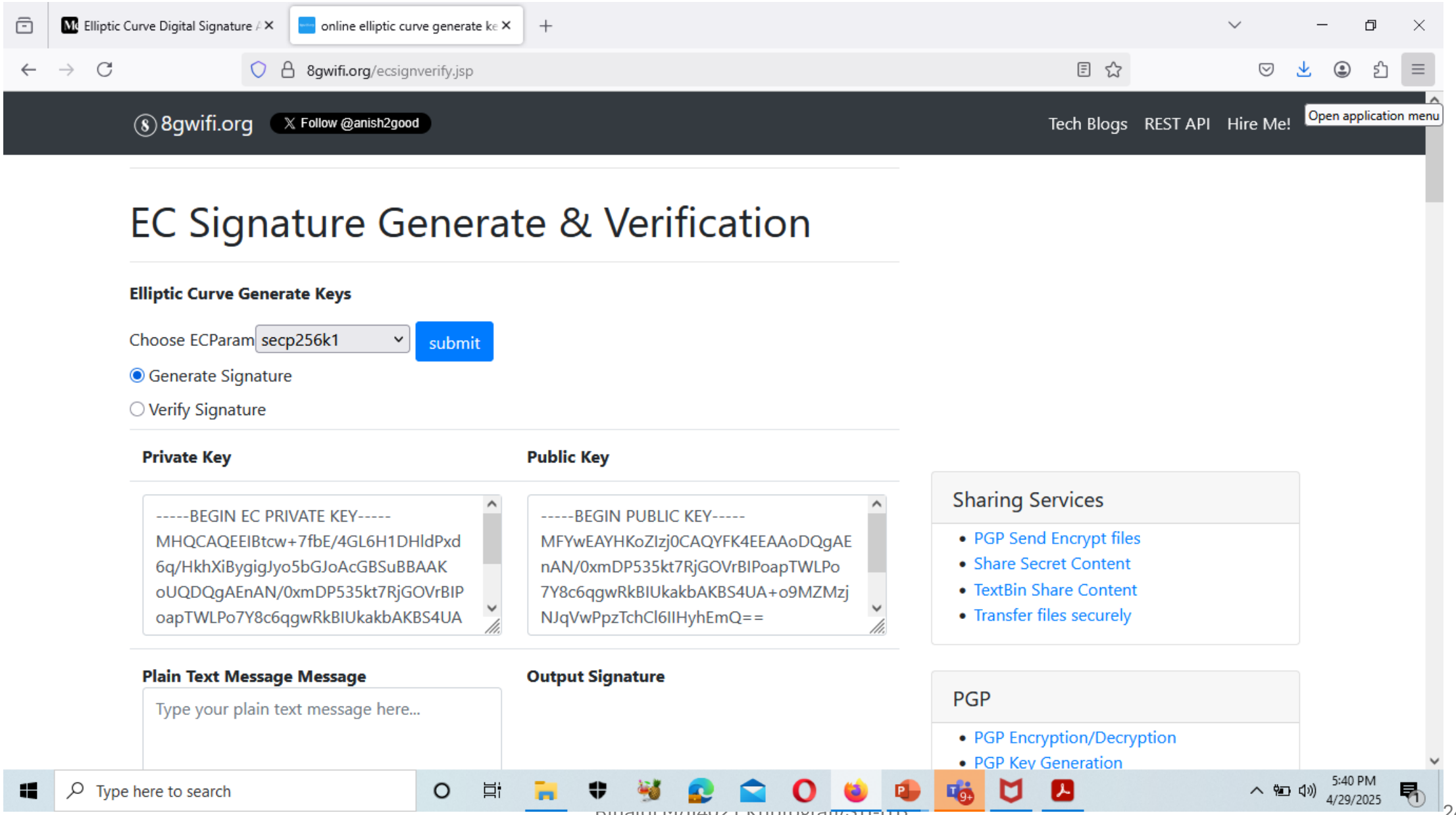
Pengirim pesan mengirim pesan m + (r, s) kepada penerima pesan.

2. Verifikasi tanda-tangan

Penerima pesan memverifikasi tanda-tangan sebagai berikut:

1. Periksa apakah r dan s terdapat di dalam selang [1, n-1]
2. Hitung e = HASH(m).
3. Hitung $w = s^{-1} \bmod n$
4. Hitung $u_1 = ew \bmod n$ dan $u_2 = rw \bmod n$.
5. Hitung titik $(x_1,y_1) = u_1G + u_2Q$.
6. Tanda-tangan valid jika $r = x_1 \bmod n$, invalid jika bukan.

https://8gwifi.org/ecsignverify.jsp



SCHNORR DIGITAL SIGNATURE SCHEME

- Seperti ElGamal, skema tanda-tangan Schnorr juga didasarkan pada logaritma diskrit
- Skema Schnorr meminimumkan komputasi yang dibutuhkan dalam pembangkitan tanda-tangan digital

- Pekerjaan utama untuk pembangkitan tanda-tangan digital tidak bergantung pada pesan dan dapat dilakukan selama waktu diam (idle) prosesor.

- Skema Schnorr didasarkan pada penggunaan sebuah bilangan prima p, yang memiliki faktor prima (p - 1) dari sebuah integer q; yaitu $p \equiv 1 \pmod{q}$. Secara khusus kita menggunakan $p = 2^{1024}$ dan $q = 2^{160}$. Jadi, p adalah bilangan 1024-bit number, dan q adalah bilangan 160-bit, yang juga adalah Panjang nilai hash SHA-1

Algoritma

Pembangkitan Kunci Publik dan Kunci Privat:

Step-1: Pilih bilangan prima p dan q, sedemikian sehingga q adalah faktor prima dari p-1, yaitu $p \equiv 1 \pmod{q}$.

Step-2: Pilih sebuah integer α, sedemikian sehingga $\alpha^q \equiv 1 \pmod{p}$. Nilai α, p, dan q adalah *global public key* yang dapat digunakan bersama di dalam kelompok pengguna.

Step-3: Pilih sebuah bilangan acak s dengan syarat $0 < s < q$. Ini adalah kunci privat pengguna.

Step-4: Hitung $v \equiv \alpha^{-s} \pmod{p}$. Ini adalah kunci publik pengguna.

Pembangkitan Tanda-tangan Digital:

Step-1: Pilih bilangan acak r dengan syarat $0 < r < q$ dan hitung $x = \alpha^r \bmod p$. Ini adalah tahap pre-processing yang independent dari pesan M yang ditandatangani.

Step-2: Sambungkan (concate) pesan M dengan x lalu hitung nilai hashnya:

$$e = H(M \,||\, x)$$

Step-3: Hitung $y = (r + se) \bmod q$. Tanda-tangan digital terdiri dari pasangan (e, y).

Verifikasi tanda-tangan digital

Step-1: Hitung x'

$x' = \alpha^{y} v^{e} \bmod p$

Step-2: Verifikasi e = H (M || x').

Di sini, H (M || x') = H (M || x).

Untuk memastikan verifikasi ini bekerja, amatilah bahwa

$x' \equiv \alpha^{y} v^{e} \equiv \alpha^{y} \alpha^{-se} \equiv \alpha^{r} \equiv x \pmod p$